SYSTEM FOR PRODUCING MEMORY MAPS BY INTERPRETING A DESCRIPTOR FILE WHICH IDENTIFIES AND DESCRIBES THE DATA STRUCTURES PRESENT IN MEMORY

This application is a continuation of application Ser. No. 913,157, filed Sept. 26, 1986, now abandoned.

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates in general to a technique for allowing programmers to generate customized displays from memory or from memory dumps, in order to carry out problem analysis of operating systems/programs with differing internal structural format.

2. Prior Art Systems

Computer operating systems characteristically use control structures to maintain order in processing. The control structures of an operating system are small collections of related information pertaining to some feature of the operating system. The system software programs that use these control structures have descriptions thereof embedded in the program code. Refer for example to the dump program referred to as Dpedit used in association with the Honeywell MOD 400 Operating System. Because these descriptions are embedded in the compiled or assembled program code, whenever improvements or alterations are made to an operating system requiring any altering or adding to the control structures, it is necessary to recompile or reassemble any programs that use those control structures that were altered. This becomes a time consuming task. Furthermore, in the past more than one version of a processing program would have to be maintained. When changes occurred to the operating system it was common to simply provide a new recompiled version of a structure-dependent dump program to customers. This meant that multiple versions of the dump program had to be maintained.

SUMMARY OF THE INVENTION

To accomplish the foregoing and other objects, features and advantages of the invention there is provided a control structures file that is adapted to provide a service to and be accessed by a dump program or routine. Information previously stored in the dump (utility) program which is variable is now accessed via the structures file. This means that control structure updating does not involve program alteration. Furthermore, the need to provide and maintain multiple versions of the utility program so as to support multiple sets of the information is eliminated. The process of using customized structure files and a single processing program allows a user to carry out problem analysis on different systems or different versions of the same system by using the same command set. The command choices available to the user are independent of the kind of internal format of the memory entity being analyzed.

In brief, in accordance with the invention there is provided a process for generating a logical display of memory values in response to commands generated by a user for any one of a number of different versions of a program employing different control structures. The process broadly comprises the steps of storing a plurality of descriptors coded for describing all of the control structures used by the program, accessing the descriptors in an order specified by the user, and interpreting the accessed descriptors to generate a logical display of the control structure data described by the descriptors for the program.

In summary one feature of the present invention is to provide, in association with the dump program, a control structures file adapted to hold the descriptions of all of the control structures, from which the processing de-bugging program can dynamically load and use the particular structure needed at any one point in time.

Another feature of the present invention is to provide a single version of processing program adapted to support all versions of control structure sets.

A further feature of the present invention is to provide a control structures file as in accordance with the preceding objects and in which the updating of a file of control structures to fit a new software release can be carried out by an automated process using the new structure templates as the input.

Still another feature of the present invention is to provide a structures file to contain control structure descriptions in a usable and easily updatable format so as to not require program recompilation when structures change.

Another feature of the present invention is to provide a structures file in which the processing program is able to access any number of these files each containing a set of descriptions of control structures applicable to one version of operating system, or alternatively sets of data structure descriptions for other operating systems.

Still another feature of the present invention is to provide a process of using customized structure files in association with a single processing program allowing a user to carry out his own specialized problem analysis on different systems or different versions of the same system using the same command set.

Still another feature of the present invention is to provide a process of using customized structure files to provide the user, unfamiliar with a particular structure, a verbal description of each field of that structure.

Another feature of the present invention is to provide a process of using customized structure files and to provide, for each structure, a hierarchical road map of how to reach that structure so that the processing program can locate and display data automatically for the user.

BRIEF DESCRIPTION OF THE DRAWINGS

Numerous other objects, features and advantages of the invention should now become apparent upon a reading of the following detailed description taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a general block diagram illustrating the broad concepts of the invention;

FIG. 2 is a diagram illustrating control structures for an operating system in the form of a tree of structures with a common point from which all structures are accessible;

FIG. 3 is a flow chart of the DISPLAY command;

FIG. 4 is a flow chart of the LOCATE command;

FIG. 5 is a flow chart of the HIER-DESC ACTION;

FIG. 6 is a flow chart of the BACK-UP function;

FIG. 7 is a flow chart of the CHAIN function;

FIG. 8 is a flow chart of the DISPLAY function;

FIG. 9 is a flow chart of the GET STRUCTURE AND DATA function;

FIG. 10A is a flow chart of the SEARCH HIERARCHY DESCRIPTION function;